NutriFlow

Features:

- Meal Planning (Nutrition/Recipes)
    - MVP
        - USER can CRUD FOOD
        - USER can CRUD MEALS (comprised of FOOD and/or INGREDIENT)
        - USER can CRUD MEALPLANS (comprised of MEALS)
        - USER can CRUD RECIPIES (instructions for a number of servings of a MEAL)
        - USER can track daily nutrition via intake log
    - Post-MVP
        - Macro/budget aware auto-planner
        - Meal swaps/suggestions
        - Seasonal rotations
- Cart Management (Procurement)
    - MVP
        - USER's CART can be populated based on MEALs planned automatically (factoring in PANTRY contents)
        - USER can manually populate cart using search feature
        - USER can modify CART contents however they see fit (add, modify, remove)
            - USER receives warning when CART doesn't match planned MEALs
        - USER can view price comparison of STOREITEMs and RECIPIEs
            - Dedicated window for bargain hunting?
        - USER can opt to view recommendations for STOREITEMs based on nutrition similarity, store location, price
        - USER's CART can be autoloaded into relevant store e-commerce platforms for purchase
    - Post-MVP
        - Autoload into store e-commerce (checkout bridges)
        - Coupons
        - Recommend Delivery/pickup slots
- Pantry Management
    - MVP
        - PANTRY maintains a list of the STOREITEM/FOODs in the USERs HOUSEHOLD

        - USER receives warnings and recommendations based on PANTRYITEMs
            - Expiry warnings (stale, freezerburn, rotten)
            - RECIPE recommendations
        - USER can scan receipt to expedite PANTRY update after a shopping trip (basic OCR)
        - USER can add or remove INGREDIENTS/FOOD in the PANTRY
    - Post-MVP
        - Barcode scanning
- Collaboration (Household)
    - MVP
        - USER can join a single HOUSEHOLD
        - USER can invite other USERs to HOUSEHOLDs
        - USER can add MEALs/MEALPLANs to the HOUSEHOLD
        - USER can subscribe to MEALPLANs of the HOUSEHOLD as well as their own custom MEALPLANs
        - PANTRY and CART for a HOUSEHOLD are based on active MEALPLANs from all member USERs
        - member USERs can modify HOUSEHOLD CART and PANTRY
        - admin USER can restrict CART/PANTRY modification privileges of other HOUSEHOLD USERs
        - USER can see the ACTIONs of other USERs on activity/action log
    - Post-MVP
        - Multiple HOUSEHOLDS per user
        - Child accounts
- Exploration (Search)
    - MVP
        - Full test search with filters for following items
        - USER can search for specific INGREDIENT, FOOD, RECIPE
        - USER can browse available RECIPIES
            - Cuisine/Cultural exploration and recommendations
    - Post-MVP
        - Semantic search (recommend other results that may not contain explicit text)
- Miscellaneous Ideas to add
    - Provide option to use calculated calories (from macros) rather than reported calories
    - Provide a manual shopping view of the cart (as an interactive shopping list), as well as in-store location information based on the store the user is shopping at
    - Integration with restaurants (scrape nutrition facts and implement meal creation modal (maybe integrate with doordash or something similar?)

DATA MODELS

TODOS:

- **Revisit Intake Log**

user

- id: UUID
- first_name: string
- last_name: string
- email: string (unique)
- password: hash
- tier: enum('free','premium')
- preferences: preferences
- households: household_membership.id[]
- active_mealplan_ids: meal_plan.id[]
- budget_monthly?: number (maybe move later)
- created_at: datetime
- updated_at: datetime
- settings: JSONB (dict<string, any>) - TBD

---

household

- id: UUID
- name: string
- admin_id: user.id
- members: household_membership.id[]
- meal_plans: meal_plan.id[]
- pantry: pantry_item[]
- carts: cart[]
- receipts: receipt_import.id[]
- preferences: preferences
- budget_monthly?: number
- created_at: datetime
- updated_at: datetime
- appliances: appliance[]

---

household_membership

- id: UUID
- household_id: UUID
- user_id: UUID
- role: enum('admin','member','viewer')
- can_modify_cart: bool
- can_modify_pantry: bool

---

appliance

- id: UUID
- type: enum('oven','stovetop','microwave','pressure_cooker','slow_cooker','air_fryer','blender','food_processor','stand_mixer','hand_mixer','toaster','toaster_oven','coffee_maker','espresso_machine','kettle','rice_cooker','grill','deep_fryer','ice_cream_maker','bread_machine','sous_vide','juicer','other')[]
- specification: appliance_specification
- created_at: datetime
- updated_at: datetime

---

appliance_specification

- capacity?: number
- capacity_unit: enum('qt','L',…)
- speed_settings?: number[]
- power_watts?: number
- pressure_levels?: number[]
- attachments?: enum('dough_hook','whisk',…)[]

---

ingredient (not-user defined, scraped from internet)

- id: UUID
- name: string
- nutrition_facts: nutrition_facts
- category: food_category (below)
- subcategory: sub_category (below)
- allergens: allergen[] (below)
- brand?: string
- image_url: string
- created_at: datetime

- updated_at: datetime

---

nutrition_facts

- serving_size: quantity[] (need a weight and a volume to calculate density to convert between units for recipes)
- density?: number
- calories: number
- protein: number
- carb: number
- fat: number
- fiber: number
- sugar: number
- micros: JSONB (dict<string, number>)

---

receipt_import

- id: UUID
- store_id?: store.id
- image_urls: string[]
- status: enum('uploaded','parsed','applied','error')
- lines: receipt_line[]

---

receipt_line

- name: string
- barcode?: barcode.id
- quantity: quantity
- price: money
- matched_store_item_id?: store_item.id
- confidence?: number

---

quantity

- amount: number
- unit: enum('g','ml','lb','oz','unit')

---

barcode

- id: UUID
- type: enum('UPC','EAN','GTIN')
- value: int

---

food (user-defined, specific to each user/household)

- id: UUID
- name: string
- owner_id: user.id
- nutrition_facts: nutrition_facts
- category: category (below)
- subcategory: sub_category (below)
- allergens: string[] (below)
- visibility: enum('private','household','public') (try to avoid data duplication by defaulting to public)
- image_url: string
- created_at: datetime
- updated_at: datetime

---

store_item

- id: UUID
- name: string
- ingredient_id: ingredient.id
- store_id: store.id
- price: money
- num_servings: number
- image_urls: string/hash[] (likely need to switch to storing actual images in the future)
- product_url?: string/hash
- in_stock: boolean
- price_history: price_point
- created_at: datetime
- updated_at: datetime

---

money

- amount: number
- currency: enum('USD','CAD',…)

---

price_point

- at: datetime
- price: money

---

store

- id: UUID
- name: string
- chain?: string
- location: geo_point
- hours: dict<day, time>
- logo_url: string/hash
- auth_token_ref: string
- created_at: datetime
- updated_at: datetime

---

geo_point

- lat: number
- lon: number

---

pantry_item

- household_id: household.id
- one of
    - store_item_id?: store_item.id
    - ingredient_id?: ingredient.id
    - food_id?: food.id
- servings_left: number
- opened: boolean
- expiration_date?: date
- storage_medium: enum('dry','refrigerated','frozen')
- created_at: datetime
- updated_at: datetime

---

cart (can include items from multiple stores)

- status: enum('open','submitted','fulfilled','canceled')
- cart_items: cart_item[]
- planned_for_range: dict<start_date: date, end_date: date>
- created_at: datetime
- updated_at: datetime

---

cart_item

- store_item_id: UUID
- num_items: number
- substitutable: bool (later updated to possible substitutes)
- notes?: string[]

---

recipe

- id: UUID
- name: string
- owner_id: user.id
- ingredients_map: JSONB (dict<ingredient.id: quantity>)
- foods_map: JSONB (dict<food.id: quantity>)
- instructions: string[]
- estimated_cook_time: number
- servings: number
- cuisine: string
- difficulty: enum('easy','medium','hard')
- applicances: appliance[]
- image_url: string/hash[]
- tags: string[] (largely TBD)
- visibility: enum('private','household','public')
- created_at: datetime
- updated_at: datetime

---

meal

- id: UUID
- name: string (default options + custom)
- owner_id: user.id
- foods_map: JSONB (dict<food.id: quantity>)
- ingredients_map: JSONB (dict<ingredient.id: quantity>)
- recipes: recipe.id[]
- notes?: string[]
- image_url: string/hash[]
- visibility: enum('private','household','public')
- created_at: datetime
- updated_at: datetime

---

meal_plan

- id: UUID
- name: string
- owner_id: user.id
- schedule: meal_plan_entry[]
- visibility: enum('private','household','public')
- created_at: datetime
- updated_at: datetime

---

meal_plan_entry

- meal_id: meal.id
- when: datetime (more general datetime, not necessarily a day of the year)
- locked: bool

---

action

- id: UUID
- user_id: user.id
- household_id?: household.id
- action_type: enum('meal_logged','recipe_added','item_purchased','pantry_updated','cart_modified','plan_changed','login','invite_sent',…)
- metadata: JSONB (dict<string, any>)
- timestamp: datetime

---

preferences

- diet: enum('omnivore','vegetarian','vegan','keto','pescatarian','paleo','mediterranean','low_carb','high_protein')
- allergens: allergen[] (below)
- disliked: (food.id | ingredient.id | recipe.id)[]
- cuisine_preferences: string[]
- goals: enum('save_money','eat_healthier','reduce_waste','high_protein','bulking','cutting','maintenance')[]
- shopping_frequency: number (in days)
- TBD (add more as the need arises)

---

intake_log

- id: UUID
- user_id: user.id
- date: datetime
- data: intake_log_item[]

intake_log_item

- metadata: string(time of day? name of meal/snack? comments/notes?)
- At least one of:
    - foods_map: JSONB (dict<food.id: quantity>)
    - ingredients_map: JSONB (dict<ingredient.id: quantity>)
    - meal: meal

Categories & Subcategories

- **meat**
    - beef
    - pork
    - lamb
    - game (venison, bison, etc.)
- **poultry**
    - chicken
    - turkey
    - duck
- **seafood**
    - fish (salmon, tuna, cod, etc.)
    - crustacean (shrimp, crab, lobster)
    - mollusk (clams, oysters, squid, scallops)
    - seaweed
- **dairy**
    - milk
    - cheese
    - yogurt
    - butter
    - cream
- **egg**
    - chicken egg
    - quail egg

    - duck egg
- **grain**
    - rice
    - wheat
    - oats
    - corn
    - barley
    - rye
    - quinoa
- **legume**
    - beans (black, kidney, pinto, etc.)
    - lentils
    - chickpeas
    - peas
    - soy products (tofu, tempeh, edamame)
- **nut**
    - almond
    - walnut
    - cashew
    - pecan
    - pistachio
    - hazelnut
    - brazil nut
    - macadamia
- **seed**
    - sesame
    - sunflower
    - chia
    - flax
    - pumpkin
- **fruit**
    - apple
    - banana
    - berry (strawberry, blueberry, raspberry)
    - citrus (orange, lemon, lime)
    - melon
    - tropical (mango, pineapple, papaya)
- **vegetable**
    - leafy green (spinach, lettuce, kale)
    - root vegetable (carrot, potato, beet)

    - cruciferous (broccoli, cauliflower, cabbage)
    - allium (onion, garlic, leek)
    - nightshade (tomato, pepper, eggplant)
- **fungi**
    - mushroom
    - truffle
- **herb/spice**
    - parsley
    - basil
    - rosemary
    - cinnamon
    - pepper
    - turmeric
- **beverage**
    - juice
    - soda
    - tea
    - coffee
    - alcohol (beer, wine, spirits)
- **condiment/sauce**
    - ketchup
    - mustard
    - mayonnaise
    - soy sauce
    - hot sauce
    - salad dressing
- **snack**
    - chips
    - candy
    - protein bar
    - popcorn
- **baked_good**
    - bread
    - pastry
    - cake
    - cookie
- **oil/fat**
    - olive oil
    - canola oil
    - butter/lard

    - shortening
- **sweetener**
    - sugar
    - honey
    - syrup (maple, corn)
    - artificial sweetener
- **processed/packaged**
    - frozen meal
    - canned good
    - ready-to-eat (instant noodles, snack packs)
- **other**
    - supplements
    - meal replacements
    - uncategorized items

Allergens

Major (Big 14: U.S. + EU)

- milk
- egg
- fish
- crustacean shellfish (shrimp, crab, lobster)
- mollusk shellfish (clams, oysters, squid, scallops)
- tree nuts (almond, walnut, cashew, pecan, pistachio, hazelnut, brazil nut, macadamia)
- peanut
- soy
- wheat
- sesame
- celery
- mustard
- sulphites (≥10 mg/kg)
- lupin

Additional Sensitivities

- gluten
- corn
- nightshade (tomato, potato, pepper, eggplant)
- caffeine
- alcohol
- artificial sweeteners (aspartame, sucralose, saccharin)

- colorants/dyes (Red 40, Yellow 5, etc.)
- MSG
- casein (milk protein)
- lactose